NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

SIMULATION TO DETERMINE THE IMPACT OF LIFE-CYCLE MANNING ON LIEUTENANTS

by

William I. Lewis, Jr.

June 2005

Thesis Advisor: Paul J. Sanchez
Second Reader: Samuel E. Buttrey

Approved for public release; distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2005	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Simulation to Determine the Impact of Life-Cycle Manning			**5. FUNDING NUMBERS**
6. AUTHOR(S) William I. Lewis, Jr.			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) Deputy Chief of Staff, Army G-1			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE** A
13. ABSTRACT (maximum 200 words) The U.S. Army has proposed a new manning strategy to reduce personnel turbulence and build strong cohesive combat units. Life-cycle manning would synchronize officer assignment with the 3-year life cycle of a Unit of Action (UA). This thesis uses simulation to examine the length of time an officer waits between graduation from the Basic Officer Leadership Course (BOLC) and assignment to a UA. The model is a discrete-event simulation based on a Java library called Simkit. This is a terminating simulation that provides the average delay lieutenants experience before unit assignment, over a 10-year period. This thesis uses robust design to evaluate both the mean performance and the variability of the system. By minimizing a quadratic loss function, optimal settings are determined that trade off some expected delay in order to achieve greater consistency. This analysis reveals that this system behaves like a queueing model in which officer accessions influence the arrival rate and the number of life-cycle units and their fill rates influence the service rates. Reducing officer accessions and the length of the life cycles while increasing the unit strength will keep the system stable and the expected delays smaller with greater consistency.			
14. SUBJECT TERMS Life-Cyle Manning, Manpower, Simulation, Simkit, Regression, Robust Design			**15. NUMBER OF PAGES** 77
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

SIMULATION TO DETERMINE THE IMPACTS OF LIFE-CYCLE MANNING ON LIEUTENANTS

William I. Lewis, Jr.
Major, United States Army
B.S., United States Military Academy, 1992

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
June 2005

Author: William I. Lewis, Jr.

Approved by: Paul J. Sanchez
Thesis Advisor

Samuel E. Buttrey
Second Reader/Co-Advisor

James N. Eagle
Chairman, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The U.S. Army has proposed a new manning strategy to reduce personnel turbulence and build strong cohesive combat units. Life-cycle manning would synchronize officer assignment with the 3-year life cycle of a Unit of Action (UA). This thesis uses simulation to examine the length of time an officer waits between graduation from the Basic Officer Leadership Course (BOLC) and assignment to a UA. The model is a discrete-event simulation based on a Java library called Simkit. This is a terminating simulation that provides the average delay lieutenants experience before unit assignment, over a 10-year period. This thesis uses robust design to evaluate both the mean performance and the variability of the system. By minimizing a quadratic loss function, optimal settings are determined that trade off some expected delay in order to achieve greater consistency. This analysis reveals that this system behaves like a queueing model in which officer accessions influence the arrival rate and the number of life-cycle units and their fill rates influence the service rates. Reducing officer accessions and the length of the life cycles while increasing the unit strength will keep the system stable and the expected delays smaller with greater consistency.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank the Deputy Chief of Staff, Army G-1, Planning and Resources Division for providing this topic. Thanks to Colonel Galing and Major Hovda for their willingness to come to the Naval Postgraduate School and present several issues that needed research by graduate students and would serve as thesis opportunities. Thanks to LTC Lukens for his initial guidance on how to approach this analysis.

Many thanks go to Professor Paul Sanchez, my advisor, for his countless hours of teaching and patience in coaching me through this process. He kept me focused and motivated throughout the entire study. Most of all, he helped me to see how all the different concepts I learned during my two years in this curriculum fit together.

I would also like to thank Professor Buttrey who helped me with his expertise in S-PLUS to perform much of the output analysis. Professor Whitaker helped me to develop my regression models and Professor Buss helped me get my simulation off the ground.

My wife, Joy, and son, Jacob, were my greatest source of encouragement. Their patience and love made this learning process so much more enjoyable. Thanks.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The Army plans to implement life-cycle manning instead of the current individual replacement system as the personnel manning strategy. The purpose of this new strategy is to reduce personnel turbulence within units and build strong and cohesive combat units. There are 43 CONUS-based self-sustaining brigade-size combat units that will operate on a 36-month life cycle. Life-cycle manning will synchronize officer assignments with the operational cycle of these units (Force Stabilization Website, 2004).

The Deputy Chief of Staff, Army G-1, is interested in how life-cycle manning will affect company-grade officer assignments. One concern is that this implementation will create large queues of officers waiting for a unit assignment. This thesis uses simulation to examine the length of time an officer waits between graduation from the Basic Officer Leadership Course (BOLC) and assignment to a Unit of Action (UA).

This model is a discrete-event simulation based on a Java library called Simkit. The model is a terminating simulation that provides output for the average delay officers experience before unit assignment over a period of ten years. This thesis uses a technique called robust design to evaluate not only the mean performance of the expected delay, but also how sensitive it is to variability in factors beyond our control. We try to gain insights about this system by developing a mathematical model that describes the relationship between input parameter settings and observed outputs. This is done by conducting multiple

simulation runs at various combinations of the input parameter settings and observing the system performance. These input parameters, also called factors, can have a range of different levels and be explicitly controlled in the simulation.

There are four design factors controlled in this model. The length of the life cycle is separated into two distinct phases, a reset phase for personnel turnover and an operational phase for training and deployment. Other factors include the number of officer accessions per year and the percentage that a life-cycle unit is filled above its authorized strength.

After running the simulation over ten years and analyzing the results, we conclude that a reduction in the length of the life cycle and the number of officer accessions, and an increase in the percentage of overfill allowed for the authorized strength will reduce the average wait time with the greatest consistency. This study uses delays in officer assignment as it sole objective. These recommendations may differ if other objectives or trade-offs are considered.

I. INTRODUCTION

A. INDIVIDUAL REPLACEMENT SYSTEM (IRS)

The U.S. Army currently uses the Individual Replacement System as a manning strategy for regular army (RA) active duty officers. This strategy assigns officers based on a combination of unique skills and qualifications, officer preferences, and the needs of the Army. The Army began using this replacement system in 1912 due to a requirement to "place large numbers of soldiers in the combat theater" (Elton, 2002). This system was "flexible and efficient" and allowed the Army to put the right number of forces in the right place quickly (AR 600-83, 1986).

When a newly commissioned Second Lieutenant (2LT) graduates from his or her Officer Basic Course (OBC), the Army assigns the officer to a temporary replacement unit at a specific installation. Upon arrival, the replacement unit assigns the officer to a major army command (MACOM). Each MACOM manages current and projected officer strength by rank and branch. As officers arrive, they are distributed to subordinate units to fill positions based on priority. A 2LT is promoted to First Lieutenant (1LT) during the first assignment. For this analysis, 2LT and 1LT positions will be considered to be synonymous; both will be referred to simply as lieutenants. Once a lieutenant has served at least 24 months in the unit, he or she is eligible for reassignment based on the needs of the Army. However, most permanent change of station (PCS) orders for officers reassign them after 36 months in the unit. For Lieutenants, the next assignment is usually attendance at the Captain's Career Course (CCC) for his or

her respective branch and is concurrent with promotion to the rank of Captain.

Subordinate units such as the battalion and company level hold large numbers of positions for lieutenants. These positions include, but are not limited to, executive officers, platoon leaders, and various staff officer positions. The number of positions available varies by branch and unit type. Traditionally, these unit types have consisted of both heavy and light force structure. Under the Army's transformation and plan of modularity, the force structure will consist of the traditional light and heavy forces as well as the addition of the Stryker Brigade Combat Team (SBCT). These changes have prompted the proposal for a new manning strategy.

B. PROPOSED CHANGE – LIFE-CYCLE MANNING (LCM)

One of the shortcomings of the IRS manning strategy is the high degree of personnel turnover within the unit. Such turbulence within a unit creates challenges when trying to build cohesive organizations prepared to work together in combat. The new life-cycle manning strategy, also known as "unit-focused stability", attempts to reduce turbulence and create strong cohesive combat units (Force Stabilization Website, 2004).

Under unit-focused stability, the Army will man 43 CONUS-based self-sustaining brigade-size combat teams known as Units of Action (UA) using a life-cycle manning strategy (Force Stabilization Website, 2004). The Military Strength Analysis and Forecasting Division of the Deputy Chief of Staff, Army G-1, is interested in how life-cycle manning will affect the availability of company-grade officers.

The life-cycle manning strategy will synchronize an officer’s assignment with the operational cycle of a UA for 36 months (Force Stabilization Website, 2004). UAs will rotate on a staggered schedule allowing up to 4 of the units to reset and refill personnel requirements each quarter. The Army will assign second lieutenants to available UAs upon graduation from phase III of the Basic Officer Leadership Course (BOLC). Once an officer completes a 36-month tour with a life-cycle unit, the officer attends a 20-week branch specific Captain’s Career Course (CCC). Upon graduation, the officer is assigned to another life-cycle unit serving in various positions to include company command. However, a certain percentage of newly commissioned officers will not serve in these life-cycle units. Instead, they will serve in units that are manned in a way similar to the traditional IRS manning strategy.

Since the Department of the Army has not finalized the policies for life-cycle manning, I will use simulation to explore various factors with the goal of offering insight to decision-makers for determining policy. The remaining chapters of this thesis will describe the model designed to represent this proposed manning strategy through simulation, the output analysis that explores the factors that are significant in the model, and the results.

THIS PAGE INTENTIONALLY LEFT BLANK

II. THE MODEL

A. PROBLEM STATEMENT

Implementing the proposed life-cycle manning strategy could cause significant personnel problems. It may create large queues of officers waiting for a unit assignment or conversely, a shortage of officers that leave unit fill rates below 100 percent. This thesis will model the flow of Regular Army (RA) lieutenants in 11 of the Army Competitive Category (ACC) branches through units that are life-cycle manned. This process can be modeled as a queueing system because lieutenants wait in a queue until assigned to a unit. Four of the ACC branches are not modeled due to their lack of involvement in the UA. These four are Aviation, Finance, Adjutant General Corps, and Air Defense Artillery branches. The 11 branches that are modeled are found in Appendix A. Using simulation, this thesis will examine the length of time an officer waits between graduation from BOLC and assignment to a UA as well as fill rates for these units.

B. EVENT GRAPHS

This model is a discrete-event simulation (DES) that uses a Java library called Simkit developed at the Naval Postgraduate School (NPS) by Professor Arnold Buss (Buss, 2001). DES is a way of modeling a system by describing how the state changes over time. The state of a system is the set of variables and their values that represents what the system looks like at any particular point in time. Events are those points in time when the system state changes. The sequence of events is managed by an event list (Law and Kelton, 2000). The event list maintains a priority queue

of pending events based on the minimum time until occurrence. When an event occurs, the event notice is removed from the event list, the state variables of the system are updated instantaneously, and further events are scheduled. "No simulated time passes when an event occurs" (Buss, 1996).

"Event graphs are a way of graphically representing discrete-event simulation models" (Buss, 1996). Professor Lee Schruben first used this concept in 1983 to explain the event logic of a model (Buss and Sanchez, 2002). Professor Buss does an excellent job explaining event graph methodology in his April 2001 paper, "Basic Event Graph Modeling" (Buss, 2001). I will provide a brief summary of his paper to help the reader better understand the event graphs used to describe my model.

Event graphs consist of two major components, nodes and edges. The nodes represent the events and the edges connect the nodes to identify how events are scheduled. The edges may have a time delay to indicate how much time is to pass before the scheduled event will occur. The edges may also place a conditional argument that must be satisfied in order for the event to be scheduled. The interpretation of Figure 1 below is as follows: "The occurrence of event A causes the scheduling of event B after some time delay **t** given condition (**i**) is true" (Buss, 2001).

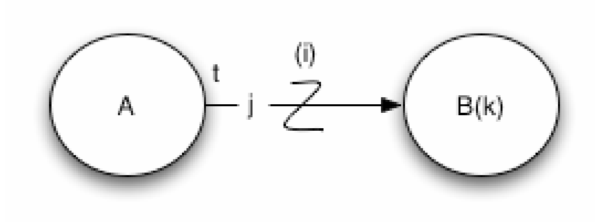

Figure 1. Basic Event Graph

The Z-shaped wavy line is a symbol that is placed on the edge to indicate that a condition must be satisfied before event B may occur. The time delay is placed at the tail of the scheduling edge and the condition is placed above the wavy line at the center of the edge. If no time delay is indicated on the graph, then the scheduled event would occur immediately. Event graphs can also pass values along the scheduling edge as parameters for the future event. In the figure above, the parameter **k** for event B will be set to the value given by **j** (Buss, 1996)**.**

The Simkit simulation library uses event graphs as the basis for building models. Each node in the event graph becomes a method. The method name is the same as the name of the node with an addition of a 'do' prefix. The names of the methods representing the nodes in Figure 1 are doA() and doB(k). The scheduling edges are implemented through the use of a waitDelay() method that has a signature with two or three arguments. The two-argument signature is waitDelay(String, double). This method schedules an event represented by the first argument for some time delay in the future represented by the second argument. The three-argument signature does the same with the addition of an object to pass as a third argument. In Figure 1, the waitDelay(String, double) method would schedule an event B after some time delay **t** and would appear as waitDelay("B",

t) in the code. Initialization is done through a 'Run' event represented as doRun() and is automatically placed on the event list at time zero.

Simkit can also link different components together through the use of a SimEventListener object. This object is used when the occurrence of an event in one simulation component should trigger the occurrence of an event with the same signature in another simulation component. For a thorough explanation of how Simkit uses the SimEventListener, see the 2002 paper, "Component Based Simulation Modeling with Simkit", by Professor Buss.

C. LIFE-CYCLE MANNING COMPONENTS

This model operates like a multiple server queue to mimic the behavior of the proposed manning strategy consisting of both life-cycle and non-life-cycle units. Officers newly commissioned into the rank of Second Lieutenant arrive into the system upon graduation from their respective OBC. According to the study sponsor in the Army G-1, a certain percentage of the officers in each branch will be assigned to life-cycle units while the remainder will be assigned to non-life-cycle units that are manned in a way similar to the current IRS manning strategy. In this model, officers assigned to life-cycle units will, upon arrival, proceed directly to units with vacancies that are executing the reset phase of their life cycle. If there are no vacancies at the time of the officer's arrival, the officer must wait in a queue associated with his or her respective branch and await the next arriving unit with vacancies for that branch. Units arrive into the system as they enter the reset phase of their life cycle. Units immediately fill their vacancies

if there are officers waiting in the queue. Once the queues are emptied or if the queues are empty at the unit's arrival, the unit may wait for the duration of the reset phase, a maximum of 90 days, for additional officers to arrive and fill their positions. If no officers arrive within the 90 days, the unit must transition to the training phase of its life cycle regardless of the unit's strength. As units conclude their 36-month life cycle and return to the reset phase, the lieutenants in the unit, most of whom will have been promoted to captain, will be reassigned to the CCC for continuing education.

This model consists of four distinct modules that can function independently, yet must interact with one another to execute this matching of officers to units. The first module, called Unit Reset, represents the multiple servers of the queue which are the life-cycle units. There are 43 brigade-size units, known as Units of Action (UA), serving as unit entities in the system. Each of these units has unique attributes. Such attributes include unit type, the date of entering the reset phase of the life cycle, and the specific number of lieutenant positions for each branch. A unit entering the reset phase of the life cycle corresponds to an arrival event in the simulation. As unit entities arrive into the system, the model attempts to fill the unit's vacancies with as many officers as are available before the unit transitions to the training phase of the life cycle.

The second module, called OBC Graduation, controls the arrival of officers into the system. There are 11 active army ACC branches that influence the system, each consisting of an OBC that produces graduates throughout the

year. The graduation dates are similar within a branch from year to year. However, they may vary among the branches. These graduation events introduce officers as entities into the simulation. Each officer has his or her own set of attributes such as a branch, a rank, and a graduation date from OBC.

The third module, called Lieutenant, "listens" to the arrival both of lieutenants that have completed OBC and of units that have entered their reset phase. This third module actually makes the assignment of the new lieutenants to either life-cycle or non-life-cycle units, accounts for attrition of officers during their first assignment and determines which officers are promoted to the rank of Captain and continue to the Captain's Career Course.

The fourth and final module of the simulation, called CCC, ends the simulation and collects information on the other three modules.

Figure 2 is a graphical representation of how these four modules interact. An arrow-shaped connecting line that resembles a stethoscope is used to indicate which component is listening. Having provided a general overview of the model, I will now discuss the individual modules in greater detail. For access to the code of this simulation, you may make requests to the author at William.lewisjr@us.army.mil.

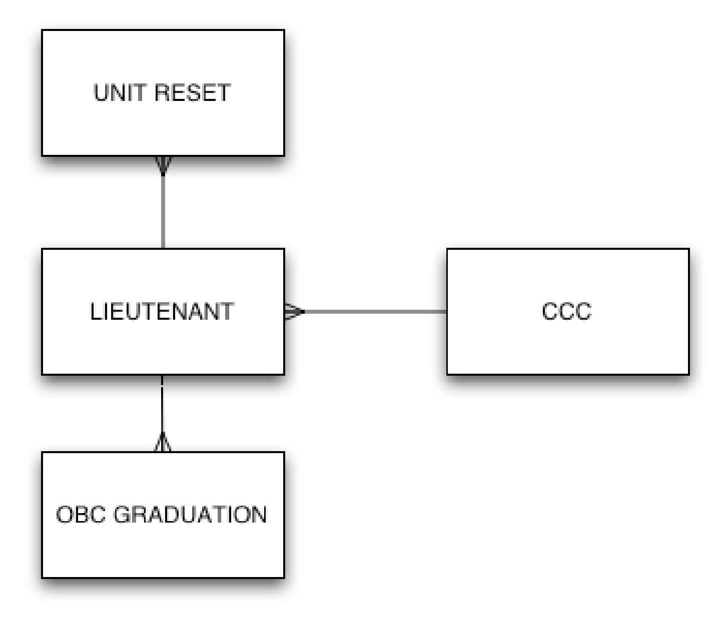

Figure 2. Simulation Modules

1. Unit Reset

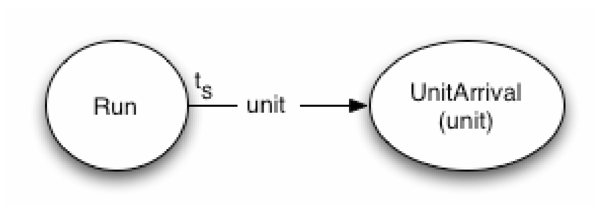

Figure 3. Unit Reset Event Graph

As illustrated in Figure 3, the simulation begins with a run event inside the Unit Reset class that initializes the simulation at time 0. The run event schedules unit arrival events to occur by invoking a method called **doRun()**. This method schedules each of the 43 units to enter its initial reset phase of the life cycle. The Military Strength Analysis and Forecasting Division of the Deputy Chief of Staff, Army G-1, provided the dates in which 38 of the units will begin its life cycle. The study sponsor also provided guidance to project the remaining 5

units to begin its life cycle several years into the future. For this analysis, I staggered the life cycles of these 5 units beginning in 2010. The units are created as one of three types: Stryker, Light, or Heavy units. The unit objects are instantiated inside the **doRun()** method and each is passed as a parameter along the scheduling edge from the run event to the unit arrival event with a time delay of t_s. The time delay parameter t_s represents the delay until the unit arrival event occurs. The **doUnitArrival()** method represents this unit arrival event and contains the logic for how the units will fill their vacancies during the reset phase of the life cycle. This method is an empty placeholder in the Unit Reset module. The Lieutenant class is the third module and is registered to listen for the occurrence of unit arrival events. When these events are heard, the matching **doUnitArrival()** method in the Lieutenant class is activated and performs the event logic.

2. Graduation

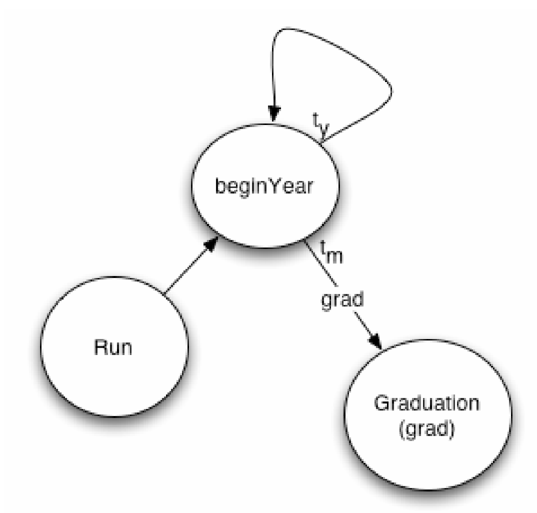

Figure 4. Graduation Event Graph

The simulation also initiates a run event inside the Graduation class. The run event schedules the first occurrence of an event called **beginYear()** that will determine the graduation sequence for the upcoming year for each of the 11 ACC branches in this model. The default behavior is to "bootstrap" by randomly sampling historical data. This is accomplished by selecting an array that contains graduation data from one of the fiscal years from among the previous eight. This historical data will serve as input data for the model. The simulation begins in fiscal year 2007 and runs for ten years. The model will randomly select one of the previous eight years to represent 2007 and repeat this process for the remainder of the simulation. For example, the data from 2003 might be randomly selected to serve as input data for 2007 for the infantry branch. In this case, the number of infantry courses that occurred in 2003 along with the number of graduates from each course would be used as input data to model the year 2007. Due to the stochastic nature of this model, the data from 1998, for example, could be randomly chosen as input data for the field artillery branch in 2007. Each branch will follow its own empirical distribution. This assumes that the accessions for the next 10 years will be similar to those of the last 8 years. As illustrated in Figure 4, the **beginYear()** event will schedule another **beginYear()** event to occur after a time delay of one year in order to generate the graduation sequence for the next year. This time delay parameter is represented by the variable, t_y. For this model, the index y for the time delay parameter is equal to one.

The **beginYear()** event instantiates a graduation which tracks the day of the year for the graduation, the number of graduates, and the branch. The **beginYear()** event schedules all graduation events for the 11 active army ACC branches in the current year. The graduation object is passed as a parameter along the scheduling edge so that the branch and number of graduates are available when the graduation event occurs. The variable t_m represents the delay from the beginning of the year until the graduation events occur. Personnel from the Army G-1 office provided course graduation dates and the corresponding number of graduates for the past eight years for each branch from the Army Training Requirements and Resources System (ATRRS).

3. Lieutenant

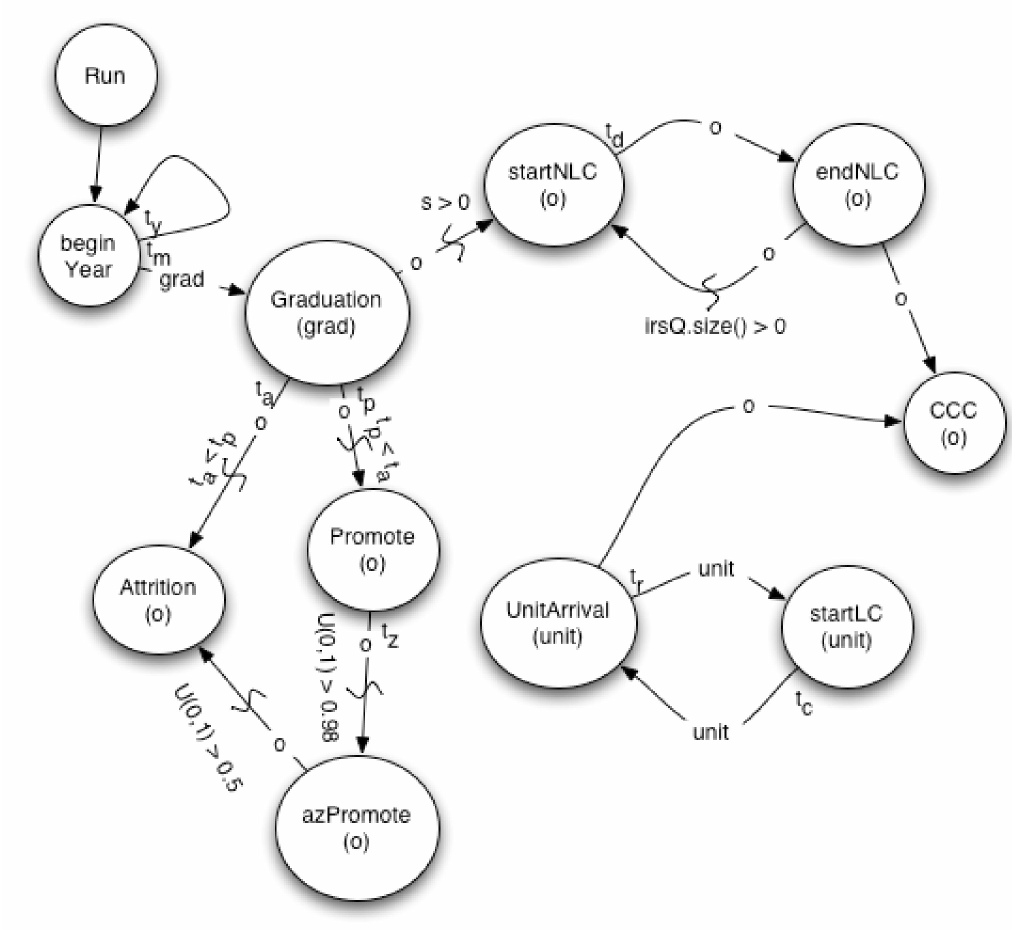

Figure 5. Lieutenant Event Graph

This class is the workhorse of the simulation. This module "listens" to the arrival of units and the OBC graduation events that occur in two of the other three modules through a linkage called a "SimEventListener". As these events occur and are heard by the listener, the corresponding methods in this class execute the event logic. The remaining paragraphs of this section will provide a detailed description of the event graph in Figure 5.

a. UnitArrival()

Beginning with the bottom right portion of Figure 5, the node representing the UnitArrival() event handles the arrival of life-cycle units in the simulation. Unit objects initially enter the reset phase of their life cycle and are treated as "arrivals" one unit at a time according to their scheduled reset date. The **doUnitArrival()** method receives the specific unit object as a parameter when the event occurs in the simulation. Upon arrival, the simulation updates a state variable called "numberOfArrivals" by incrementing the count by one. The unit object captures the current time as its reset entry time and immediately schedules an event to "start" its life cycle after a time delay of 90 days. This time delay is a constant in the model represented by the variable t_r in the event graph above. The "start" event represents the unit's transition from the reset phase into the training phase of the life cycle and is depicted in the simulation as the **doStartLC()** method. Before making this transition, the unit attempts to fill its vacant positions with available lieutenants.

The number of vacancies filled is contingent on the number of lieutenants available for assignment during the reset phase. At the beginning of the simulation, units will arrive empty and immediately proceed to fill with available lieutenants. However, as the units complete their life cycle and return to the reset phase, they will need to remove the lieutenants who have been assigned to the unit for the past three years and have recently been promoted to the rank of captain. This removal process actually schedules a Captain's Career Course (CCC) event

and passes each removed officer as a parameter along the scheduling edge to the CCC event with no time delay. After removing these officers, the unit will refill the positions with new lieutenants that have completed OBC.

The unit will make its vacancies known by registering each vacancy by branch type. A unit with vacancies is placed into a container called UnitMap. As units arrive, they register their vacancies by placing the quantity by branch into a set inside the container. As vacancies are filled in the unit, the set removes the vacancy and decrements the count for the unit's demand by the number of vacancies filled. When the unit has filled its authorized positions and no longer has vacancies, it removes itself from the container. After 90 days, the units transition into the training phase of their life cycle represented by a **startLC()** event.

b. StartLC()

This event is driven by the number of days a unit has been in the reset phase and not by the number of positions the unit has filled. In fact, units may transition without achieving their authorized strength. The **doStartLC()** method executes this event and initially schedules another UnitArrival event for the unit to enter the reset phase again after a time delay of 3 years. This time delay variable is represented by t_c in Figure 5.

c. Graduation()

Graduation events are scheduled by the beginYear events as described in the Graduation class. Each graduation object passed into the **doGraduation()** method has an instance variable specifying the number of graduates for that course. This number of graduates determines the

number of officer objects to instantiate for the graduation event. Each branch specifies what percentage of the officers will serve in life-cycle units and non-life-cycle units. These newly instantiated officer objects are randomly allocated to life-cycle and non-life-cycle units based on that guidance.

Officers appointed to a life-cycle unit are immediately assigned to a unit with vacancies that is in the reset phase. If there are no units available, the officer is placed in a branch-specific queue to await the next arriving unit. The officers are removed from the queue in a "first-in, first-out" (FIFO) order. Each unit has an authorized number of positions for lieutenants based on branch. During the reset phase, the unit fills its branch-specific vacancies by removing the officers from their respective branch queue. If the queue is empty, the unit has the duration of its reset phase to wait for additional officers to graduate from OBC and fill the vacancies. If not enough officers arrive during this time period, the unit must transition to the training phase without filling all the vacancies.

The authorized number of lieutenant positions in a unit is maintained by a set of values for each branch. These sets are placed inside a container called unitHashMap. The unit may continue to add officers until the size of the set equals the maximum authorized value. A scalar value is used as an input setting to increase or decrease this maximum to experiment with overfill or shortages as desired.

Officers are created upon arrival of every graduation event. The **doGraduation()** method executes the

graduation events that occur on the event list. This method receives a graduation object as a parameter that determines the branch of the graduates and the number of graduates for this event. The number and branch type of the graduates corresponds to the number and branch type of the officers that are instantiated and enter the simulation.

As described earlier, the number of graduates for each graduation event is determined by bootstrapping historical data from the past eight years. As each officer object is instantiated, either a promotion event or an attrition event is scheduled for that officer based on the minimum time of the two possibilities. Since lieutenants are promoted to captain after 38 months of service and the average length of time for the OBC for each branch is 5 months, the time until promotion is 33 months. The time until an attrition event occurs is generated as a nonhomogeneous Poisson process due to the attrition rate for lieutenants being a function of time in service. If the time until attrition for the officer is less than the time to promote the officer, an attrition event is scheduled for the officer; otherwise a promotion event is scheduled.

d. Promote()

The **doPromote()** method executes the promotion events. According to historical data provided by the Army G-1, the promotion rate to captain has been at least 98 percent since 1996 with one exception. In 2004, the promotion rate was 92 percent due to the Army promoting the "best" qualified officers instead of the normal procedure of promoting "fully" qualified officers. For the purpose

of this analysis, the simulation will use a promotion rate of 98 percent. The model tracks how many officers have been promoted. Officers not promoted when first eligible have an opportunity to be promoted above-the-zone. The promotion event schedules an above-the-zone promotion event as a second promotion event called AzPromote one year in the future. The time delay variable t_z represents the constant delay of one year.

e. AzPromote()

The **doAzPromote()** event represents the above-the-zone process for officers that have been passed over for promotion to captain one time. Historical above-the-zone promotion rates to captain average about 50 percent over the past seven years. If the officer is not promoted to captain on the second opportunity, they are immediately removed from service by scheduling an attrition event that occurs with no delay.

f. Attrition()

Attrition of lieutenants in this simulation is a nonhomogeneous Poisson process in that the attrition rate for officers leaving the service is a function of time in which $\lambda(t)$ is the attrition rate of officers at time $\boldsymbol{t}$. Historical data indicates that the rate of attrition for officers depends on the number of years of service of the officer as illustrated in Figure 6. The attrition rates are small for officers with 1 to 3 years of service, but increase around the 4^{th} and 5^{th} year of service. Since the nonhomogeneous Poisson process, also called a nonstationary Poisson process, does not have the requirement for stationary increments, this seems to be a plausible technique for determining the attrition rate for officers (Ross, 2000).

At the time each officer is instantiated, an attrition event is scheduled for the officer if the calculated time until attrition is the minimum of an attrition event and a promotion event. The interval of time considered for attrition in this simulation is 4 years since lieutenants are generally promoted to captain after 38 months of service.

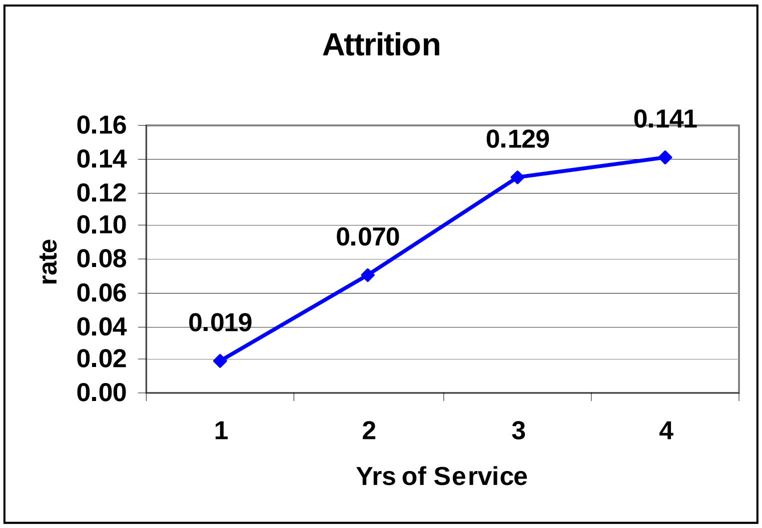

Figure 6. Officer attrition rate as a function of time

After a promotion event or an attrition event is scheduled for the created officer, the officer is randomly assigned to either a life-cycle unit or a non-life cycle unit. This assignment is branch-dependent. Each branch has an authorized number of positions in life-cycle units and a goal for the number of positions in non-life-cycle units that should be filled. Each branch has different requirements. For example, there are approximately 1,583 infantry lieutenant positions in the Army. Approximately 167 (11 percent) of those are designated as positions in non-life-cycle units that should be filled. In this model, an infantry officer is placed into one of the non-life-

cycle unit positions with probability 0.11. Otherwise they are assigned to a life-cycle unit.

Once an officer is assigned to a life-cycle unit, the officer attempts to fill a vacancy of one of the waiting units in their reset phase. If there is a unit with vacancies still in the reset phase, the officer fills this position; otherwise the officer goes into a branch-specific queue to wait for additional units with vacancies to arrive. As units arrive into their reset phase, they register their need to fill officer positions by branch type. They check each of the branch-specific queues to see if there are officers waiting. If there are officers waiting, the officer is removed from the queue and placed into the unit.

After 90 days, the units transition into the training phase of their life cycle represented by a "startLC" event. At this point, the unit unregisters its demand for vacancies with all lists in the unitMap and calculates the number of positions not filled by branch type in the unit. The unit will then continue its 36-month life cycle before returning to the reset phase to begin the personnel turnover process.

g. StartNLC()

Officers that are assigned to non-life-cycle units follow a similar pattern in that they serve a period of 3 years in the unit before reassignment to the CCC. The difference is that officers will arrive and depart these units at different times rather than all at once as with the life-cycle units.

The percentage of non-life-cycle positions for each branch determines the number of positions available

for filling by officers. These positions function like the number of available servers in a multiple server queue. In order for an officer to get assigned to a position, one must be available. Initially, all officers designated for non-life-cycle units enter a queue to wait for an available server. This queue is named "irsQ" because officers await assignment in a manner resembling the current individual replacement system. As "servers" or unit positions become available, officers are removed from "irsQ" and placed into the available position in the non-life-cycle unit to begin "service" or begin their assignment. This event schedules an "end service" event called "endNLC" after a time delay of 3 years. The time delay parameter is denoted t_d in the event graph shown in Figure 5. The officer is passed as a parameter to the **doEndNLC()** method.

h. EndNLC()

This event removes the officers from the non-life-cycle unit, schedules a CCC event and passes the officer as a parameter to the **doCCC()** method. Once the officer has been removed, a position in the unit becomes available. The model checks the size of "irsQ" to see if there are any officers waiting. If there are officers in "irsQ," the first officer waiting is removed and placed into the available position. This entire process is managed by branch. Each unit annotates its positions by branch type. When a position becomes available, the model checks "irsQ" for that branch to determine if an officer is available.

4. Captain's Career Course

This is the final event of the simulation and is provided in this model for follow-on work that would

include the CCC and assignment of captains to life-cycle units.

D. SELECTING INPUT PROBABILITY DISTRIBUTIONS

This model is a discrete-event simulation that models the proposed life-cycle manning strategy by using random inputs to determine the number of lieutenants that enter the system. When a graduation event occurs, some random number of officer objects are instantiated to represent the graduates of the course that are now ready for unit assignment. This instantiation of officer objects is how lieutenants enter the system. The issue is how many officer objects to create for each graduation event. Since this is a random variable, I examined the historical data of the number of graduates by course date in order to fit a probability distribution to the data. Due to the extensive draw-down in the strength of the Army during the early 1990's, I did not consider data prior to 1996. Parametric distribution fitting, however, requires large amounts of data. Because of the limited amount of historical data, I was unable to find parametric probability distributions that would fit the data. I resolved the problem by using the historical data as empirical distributions for the input parameters.

The total number of accessions is approximately 4500 each year and is not projected to increase significantly above this level in the near future. The technique for providing input values is called "bootstrapping" in which a single year of data is randomly selected from one of the eight previous years. These selected data will model the current year for the number of graduations and the number of graduates. This process is repeated each year until the

simulation terminates at ten years. The average number of graduates coming from the 11 ACC branches modeled in this simulation each year was approximately 3,400 during the past eight years.

E. SIMULATION VALIDATION

The study sponsor from the Military Strength Analysis and Forecasting Division of the Deputy Chief of Staff, Army G-1, reviewed and approved the event graphs and event logic of this model for validation.

F. ASSUMPTIONS

1. Accessions during the next 10 years will be similar in size to the previous 8 years at approximately 4,500 officers per year.
2. This model assumes that the average length for the officer basic course for all branches will remain 20 weeks for the next 10 years.
3. This model assumes that the promotion rate for lieutenants in their primary zone will remain at 98 percent for the next 10 years.

THIS PAGE INTENTIONALLY LEFT BLANK

III. EXPERIMENTAL DESIGN

A. ROBUST DESIGN PROCESS

This thesis uses the technique of robust design to evaluate the performance of the manpower model. Robust design is a technique that not only evaluates the mean performance of the system, but considers how sensitive the system is to uncontrollable factors that introduce variability. This is accomplished through a loss function. We use a quadratic loss function as a "surrogate for the 'true' underlying loss function which may be difficult or impossible to specify exactly" (Sanchez et al., 1996). The definition for a quadratic loss function is $L(x) = c(Y(x) - \tau)^2$ where *Y(x)* is the performance of the system as a function of an *x* vector of parameters and *c* is a scaling constant to express loss in dollars. By taking the expectation of both sides of this equation, we derive the following expression.

$$\begin{aligned}
&E[\hat{L}(x)] = cE[(Y(x) - \tau)^2] \\
&= cE[(Y(x))^2 - 2Y(x)\tau + \tau^2] \\
&= c(E[Y(x)^2]) - 2\tau E[Y(x)] + \tau^2 \\
&= c(\sigma_Y^2 + E[Y(x)]^2) - 2\tau\mu_Y + \tau^2) \\
&= c(\sigma_Y^2 + (\mu_Y^2 - 2\mu_Y\tau - \tau^2)) \\
&= c(\sigma_Y^2 + (\mu_Y - \tau)^2)
\end{aligned}$$

In other words, the expected loss is composed of the squared difference of the mean performance from our target value τ plus the variability contributed by uncontrollable factors. The variability is estimated using the square of the standard deviation regression equation (SRE) and the mean performance is estimated using the mean regression equation (MRE).

The estimated loss function is $\hat{L}(x) = (\hat{\sigma}_Y^2 + (\hat{\mu}_Y - \tau)^2)$. Ideally, we would like the average delay in the queue for this system to equal the target value τ and the variance to equal zero (Sanchez et al., 1996). This, however, is unlikely. Therefore we find controllable factor settings to minimize the estimated loss. This may involve a trade-off between mean performance and variability. For example, this technique might suggest that the best performance of the system occurs at a design point with a slightly less desirable mean performance, but with much smaller variability. For a thorough description of robust design, see "Effective Engineering Design through Simulation" (Sanchez et al., 1996).

The discrete-event simulation built in this thesis was designed to model the behavior of the proposed life-cycle manning strategy. The objective is to gain insights about the system. We try to accomplish this by developing a mathematical model that describes the relationship between input parameter settings and observed outputs. This is done by conducting multiple simulation runs at various combinations of the input parameter settings and observing the system performance. These input parameters, also called factors, can have a range of different levels and be explicitly controlled in the simulation. It is also important to vary combinations of factors jointly to see if there are interactions, also known as synergy. We use experimental designs called Nearly Orthogonal Latin Hypercubes to achieve our objective in far fewer runs than would be required using a straight combinatorial approach (Kleijnen et al., 2005). The resulting set of factor

settings and outcomes can be analyzed using the statistical technique of linear regression. Regression modeling quantifies the relationship between input settings and observed outputs. This allows us to find a combination of the factor level settings that result in the best performance of the system.

After completing the verification and validation of my model, I selected as the measure of performance the average delay in queue for a lieutenant waiting for his or her first life-cycle unit assignment. The next step in the process was to create the experiment and identify factors that could be controlled during the simulation.

B. EXPERIMENTAL FACTORS AND LEVELS

I designed an experiment to explore four controllable factors which may affect the measure of performance. I also included an uncontrollable factor that introduces variability into the model. The number of lieutenants who graduate from each OBC every year is unknown and considered a noise factor. This experiment will be a crossed design of the controllable factors with the noise factor.

The remainder of this section will specify the different controllable factors for this model and their levels. Table 1 below shows the four controllable factors in this design along with the low and high settings. The following paragraphs explain how the levels of these four factors were set.

Factor	Low Setting	High Setting
Reset Phase Length (months)	2	4
Overfill Percent	1.0	1.2
Life Cycle Length (months)	30	36
Accessions Target (Officers)	4300	4700

Table 1. Experiment Factors and Levels

1. Length of Life-Cycle Reset Phase

The current amount of time a unit will spend in the reset phase of the life cycle is 90 days. This is a turbulent time for the unit as a significant turnover of personnel occurs. The design uses a low setting of 2 months to represent 60 days and a high setting of 4 months to represent 120 days.

2. Percent to Overfill the Unit

The normal procedure is to attempt to fill units to 100 percent of their authorized strength. However, since lieutenants account for the largest proportion of the officer corps among the different officer ranks, a common practice is to overfill units with lieutenants. I received information from personnel in the Army G-1 office at the Human Resource Command (HRC) that 120 percent is the highest level for this particular practice with lieutenants. This model uses a low setting of 1.0 which

fills the units to 100 percent and a high setting of 1.2 which attempts to fill the units to 120 percent of the authorized strength.

3. Length of the Unit Life Cycle

The current length of the unit life cycle is 36 months. The first three months of the life cycle is called the reset phase and is treated separately in this experiment. This design explores the impact of varying the remaining 33 months of the life cycle. The low setting for the design is 30 months and the high setting is 36 months.

4. Target Number of Accessions

The target number of accessions for all of the ACC branches is 4500 officers each year. This target can be adjusted to allow more or fewer accessions yearly. This design explores a low setting of 4300 officers and a high setting of 4700 officers. The 4500 target value includes four branches not considered in this model. The total accessions for the 11 ACC branches considered in this model are approximately 3400. For the purpose of this analysis, I assume that an increase or decrease in the target number of accessions will be equally applied across all branches. Therefore if, for example, the target value is reduced to 4300 officer accessions, a scalar adjustment of 4300/4500 = 0.96 will be applied equally to all branches to reduce their total number of accessions.

C. DESIGN OF EXPERIMENT

A Nearly Orthogonal Latin Hypercube (NOLH) is a method of specifying factor settings to build a design matrix that ensures that the column vectors are nearly orthogonal and therefore uncorrelated (Cioppa, 2002). These column vectors become the explanatory variables of the regression equation. When the explanatory variables of a regression

equation are correlated, multicollinearity is said to exist, making the normal interpretation of the regression coefficients misleading (Neter, 1996). The purpose of this design is to reduce the correlations among the columns of the design matrix so that the relationship between the explanatory variables and the response variable can be more clearly seen. The NOLH design ensures that the estimated regression coefficients are nearly independent (Cioppa, 2002).

I used a tool developed by Professor Susan Sanchez at the Naval Postgraduate School to generate the NOLH design matrix (Kleijnen et al., 2005). This tool is an Excel worksheet that requires the user to input the low and high settings for each factor considered in the experiment. The result is a design matrix of points in which the column vectors are nearly orthogonal.

Using the NOLH sampling procedure with four two-level design factors and one three-level noise factor, the design matrix consisted of 51 design points. Due to the stochastic nature of this terminating simulation, I conducted ten replications of each design point. The result was 510 runs of the model.

D. REGRESSION EQUATION

The factors previously described were used to explore the mean response and the standard deviation response for the average delay a lieutenant experiences before his or her initial life-cycle unit assignment. Multiple regression was used to construct two metamodels for these responses (Sanchez, 1996). The following two regression equations involve a linear combination of the explanatory variables and the estimated regression coefficients. The

regression equation used to describe these two metamodels is shown below.

$$E[Y] = \beta_0 + \sum_{i=1}^{4} \beta_i x_i + \sum_{i=j}^{4} \sum_{j=1}^{4} \beta_{ij} x_i x_j$$

The response variables for the two metamodels are *Y* = average delay in queue, and *Y* = standard deviation of the average delay in queue. Although the estimated regression coefficients will be different for these two metamodels, the explanatory variables listed below will remain the same for both.

X_1=Length of reset phase

X_2=Percentage to overfill the unit

X_3=Target value for accessions

X_4=Length of the life cycle

Both metamodels initially included the main effects, all two-way interactions and quadratic effects. I used two packages of statistical software called JMP® (SAS Institute, 2003) and S-PLUS® 6.2 (Insightful Corporation, 2001) to construct the two metamodels. After examining the resulting models, I removed factors that were not statistically significant. The threshold for statistical significance used in this experiment is a p-value less than or equal to 0.05. Any main factors which did not have a p-value less than or equal to 0.05 were removed from the model unless they were included in an interaction term that remained in the model.

E. CONDUCT THE EXPERIMENT

With the design matrix complete, I ran the simulation to determine the average delay in queue for each design point. I then computed the mean and standard deviation for

the average delay in queue for each design point averaged across the noise space. For example, a design point was replicated ten times for each of the three levels of the noise factor. Each design point was averaged over the three levels of the noise factor. The result was a mean response for each of the ten replications of the 17 original design points. The standard deviation response was calculated in the same manner except the statistic collected was the standard deviation instead of the mean. The result of this process was 170 mean response and standard deviation response variables for the associated design points. Regression provided initial metamodels for each of the two responses as a starting point for analysis. The next step was to analyze the results and refine the metamodels in order to minimize the loss function.

IV. OUTPUT ANALYSIS OF EXPERIMENT RESULTS

Polynomial regression models are commonly used to approximate the true function when the behavior of a nonlinear response function is unknown (Neter et al., 1996). Knowing little about the true response function for this system, I used the method of least squares regression to fit polynomial regression models for the mean and standard deviation of the average delay in queue for this system. However, these models provided poor predictive power when the optimal design points were actually tested through simulation. Consequently, I reverted to the theory of queueing models. An M/M/1 queueing model is one in which the interarrival-times and service times follow an exponential distribution with rates λ and μ respectively, and the number of servers equals one (Ross, 2000). The queueing model used in this thesis is one in which the interarrival times and service times follow some unknown distributions and there are multiple servers. This is known as a G/G/k queueing model (Ross, 2000). This notation defines the distribution of the interarrival times and service times as generic and the number of multiple servers as some value k greater than one. M/M/1 queueing models are analytically functions of traffic intensity, $\frac{\lambda}{\mu}$, and it is reasonable to assume that G/G/k queueing systems would have behaviors which are proportional to traffic intensity. I found a reparameterization of the design factors discussed in chapter 3 which enabled me to convert these to λ and μ. I used these reparamterizations to

construct two metamodels that behaved more reasonably within the region of interest.

A. MEAN REGRESSION ANALYSIS

The first of the two metamodels is a second-order polynomial regression model for the mean response that includes main effects and interaction terms.

$$Y = average\,delay\,in\,queue$$

$$E[Y] = \beta_0 + \sum_{j=1}^{4} \beta_j x_j + \sum_{i=j}^{4} \sum_{j=1}^{4} \beta_{ij} x_i x_j$$

$\hat{\beta}_0$ is the estimated regression coefficient for the intercept term. The estimated regression coefficients for the main effects are indicated by $\hat{\beta}_j$ for j=1,2,3,4. The estimated regression coefficients for the interaction and quadratic terms are indicated by $\hat{\beta}_{ij}$ for i,j=1,2,3,4. The following are the predictor variables for this model.

X_1=Length of reset phase

X_2=Percentage to overfill the unit

X_3=Target value for accessions

X_4=Length of the life cycle

In developing the polynomial regression function for the mean response of the system, I constructed a scatter plot of the predictor variables against the mean response. This initial scatter plot revealed a region in which the variable for overfill had a much higher mean response for values less than or equal to one than for those strictly greater than one. The design points in this region stand out for each of the predictor variables plotted against the mean response. The scatter plot in Figure 7 illustrates the relationship of the predictor variables against the

mean response variable using 170 design points from the initial experiment.

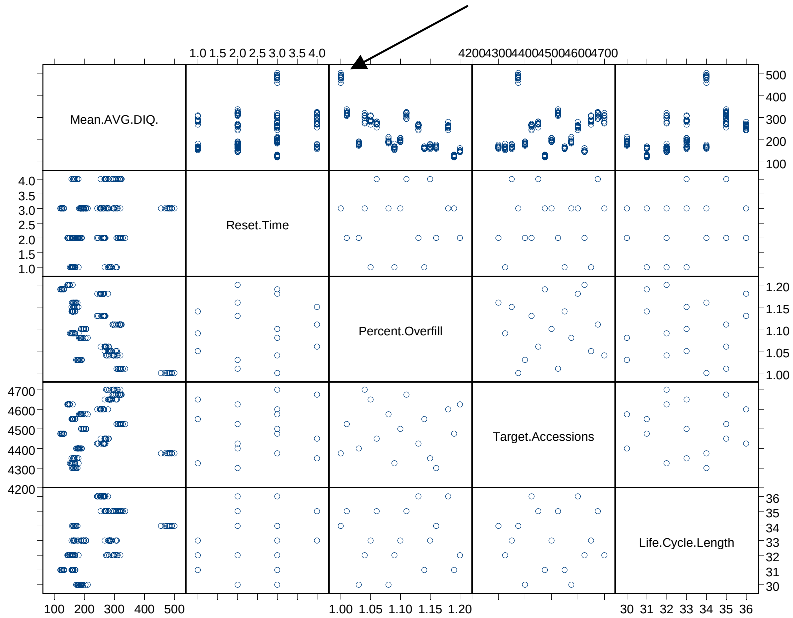

Figure 7. Original Scatterplots of Predictor Variables versus Mean Average Delay in Queue with 170 observations

The design point of interest has a setting of 3 months for reset time, 1 for overfill (representing 100 percent of the authorized strength for the unit), a yearly accessions target of 4375 lieutenants, and 34 months for the length of the training and deployment phase of the life cycle. I explored this region by conducting the design of experiment again with the same factors, but narrowed the various levels of each factor around this design point. The range of the reset time was reduced to a low setting of 2.5 and a high setting of 3.5. The range of overfill was reduced to a low setting of .95 and a high setting of 1.05. The range of target accessions was reduced to a low setting of 4350 officers and a high setting of 4400 officers. The range of the life cycle length was reduced to a low setting of 33 months and a high setting of 35 months. This provided 170 additional independent observations that I added to the

original 170 observations. Figure 8 illustrates a scatter plot of the 340 observations.

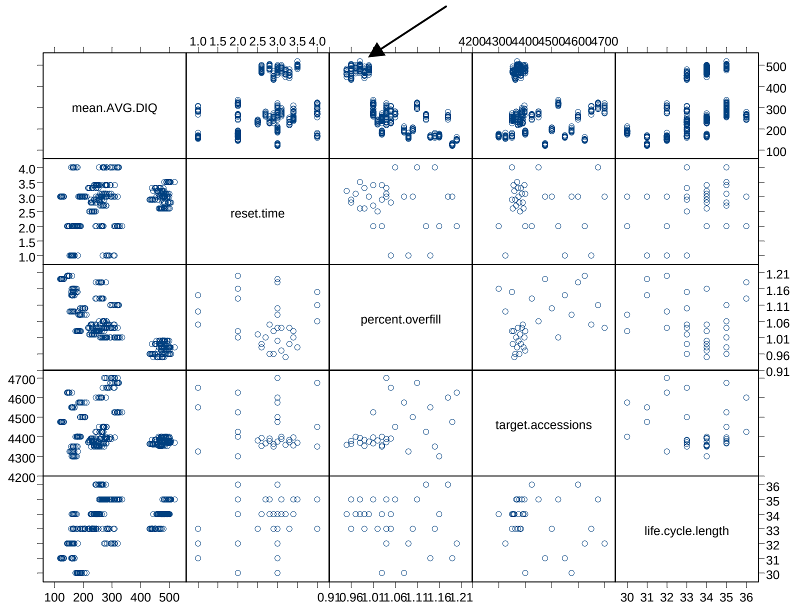

Figure 8. Appended Scatterplots of Predictor Variables versus Mean Average Delay in Queue with 340 observations

From Figure 8, it appears that the predictor variable overfill has a lower effect on the mean response variable at values strictly higher than 1.0. Values of overfill less than or equal to 1.0 correspond with much higher values of the mean response variable. Figure 9 illustrates this single factor plotted against the response variable. To capture this behavior in the model, I created an indicator variable, X_5, assigning 1 for overfill strictly greater than 1.0 and 0 for overfill less than or equal to 1.0. The revised polynomial regression model includes additional indices to represent that term and its interactions.

$$Y = average\,delay\,in\,queue$$

$$E[Y] = \beta_0 + \sum_{j=1}^{5} \beta_j x_j + \sum_{i=j}^{5} \sum_{j=1}^{5} \beta_{ij} x_i x_j$$

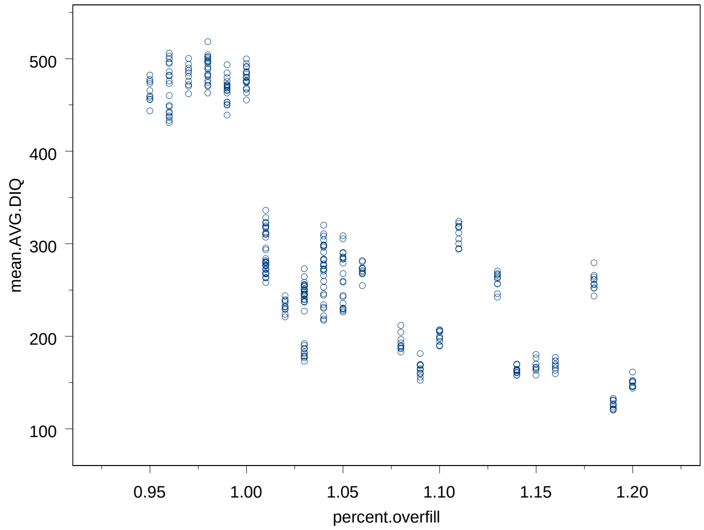

Figure 9. Scatter plot of percent overfill versus the mean response

The measure of performance for this analysis is W_Q, the expected time an officer spends waiting in queue and is a fundamental quantity in queueing models (Ross, 2000). The resulting polynomial regression function, after removing insignificant terms, for the mean delay in queue is shown below with the output from S-PLUS in displayed in Table 2.

$$\hat{W}_Q = 17422.90 + 446.74x_1 + 7269.17x_2 - 7.27x_3 - 360.65x_4 + 319.44x_5 - 118.60x_1x_2 - 0.05x_1x_3 - 3.66x_1x_4 + 17.42x_1x_5 - 0.72x_2x_3 - 39.35x_2x_4 - 590.51x_2x_5 + 0.05x_3x_4 - 1115.39x_2^2 + 0.0007x_3^2 + 2.91x_4^2$$

Coefficients:

	Value	Std. Error	t value	Pr(>|t|)
(Intercept)	17422.8956	3121.2441	5.5820	0.0000
reset	446.7425	101.3481	4.4080	0.0000
overfill	7269.1708	1056.8872	6.8779	0.0000
accessions	-7.2648	1.2124	-5.9920	0.0000
length	-360.6510	59.5603	-6.0552	0.0000
x5	319.4430	97.7759	3.2671	0.0012
I(overfill^2)	-1115.3923	361.1852	-3.0881	0.0022
I(accessions^2)	0.0007	0.0001	7.0787	0.0000
I(length^2)	2.9061	0.3804	7.6401	0.0000
reset:overfill	-118.6028	36.2316	-3.2735	0.0012
reset:accessions	-0.0470	0.0105	-4.4562	0.0000
reset:length	-3.6562	1.0272	-3.5592	0.0004
reset:x5	17.4236	5.7249	3.0435	0.0025
overfill:accessions	-0.7177	0.1495	-4.7989	0.0000
overfill:length	-39.3461	11.2836	-3.4870	0.0006
overfill:x5	-590.5133	95.0373	-6.2135	0.0000
accessions:length	0.0538	0.0115	4.6895	0.0000

Residual standard error: 10.52 on 323 degrees of freedom
Multiple R-Squared: 0.9928

F-statistic: 2782 on 16 and 323 degrees of freedom, the p-value is 0

Table 2. Mean Regression Results for Significant Terms

Figure 10 depicts a plot of the residuals versus the fitted polynomial regression model. However, this model had poor predictive power when actually tested through simulation. My next attempt to find a model with goodness of fit and better predictive power was to examine the theory of queuing models.

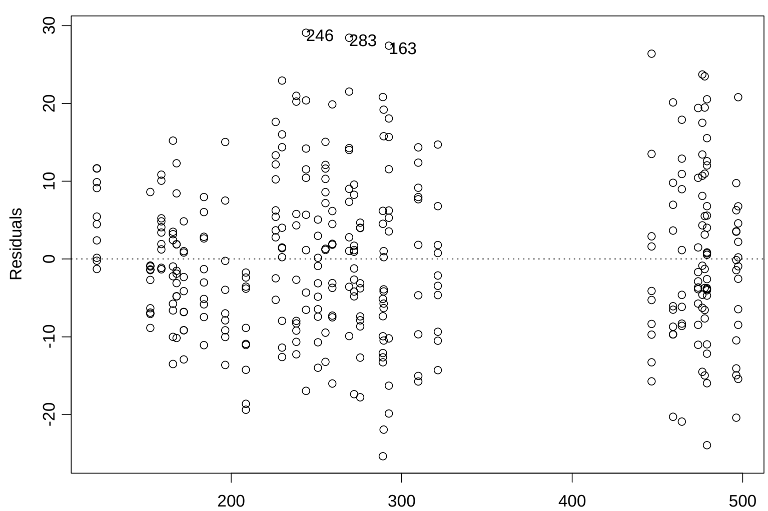

Figure 10. Plot of Residuals versus the Fitted Values

For an M/M/1 queueing system, $W_Q = \frac{\lambda}{\mu(\mu-\lambda)} = \frac{\rho}{\mu(1-\rho)}$ where $\rho = \frac{\lambda}{\mu}$ (Ross, 2000). It is plausible that delay in queue for our model should also be a function of traffic intensity. A reparameterization of the design factors in terms of λ and μ is shown below.

$$\lambda = \frac{X_3}{4500} * 3400 * (.48) = 0.3627 X_3$$
$$\mu = \frac{5072}{3} * X_2 * \frac{36}{X_1 + X_4} = 60864 \frac{X_2}{X_1 + X_4}$$

The values for λ and μ are functions of the design factors in my experiment and describe the arrival rate and service rate respectively. The arrival rate is derived by scaling the expected number of accessions that are assigned to life-cycle units per year by $X_3/4500$. The expected number of accessions per year for this model is 3400 and only 48 percent will be assigned to life-cycle units. The service rate is derived by calculating the total number of authorized positions among all the 43 life-cycle units that can be filled in a year and scaling this by some overfill value X_2. Among the 43 life-cycle units, there are 5072 positions that can service lieutenants for the length of the life-cycle. The sum of X_1 and X_4 determine this life-cycle length. A categorical variable z will represent the categorical variable X_5 from the original model. Taking the natural logarithm of W_Q transforms the equation from a quotient to a difference of the functions of the predictor variables.

$$\ln(W_Q) = \ln(\rho) - \ln(\mu) - \ln(1-\rho)$$

Since our model is not an M/M/1 system, we built an expanded model of the following form.

$$E[\ln(W_Q)] = \beta_0 + \beta_1 \ln(\rho) - \beta_2 \ln(\mu) - \beta_3 \ln(1-\rho) - \beta_4 z +$$
$$\beta_{11}(\ln(\rho))^2 + \beta_{22}(\ln(\mu))^2 + \beta_{33}(\ln(1-\rho))^2$$

The new regression equation is fitted with the predictor variables ρ, $1-\rho$, μ, and z as functions of the design factors. Since values of $\rho > 1$ create an unstable queueing system (Ross, 2000), forty design points for which $\rho > 1$ were removed from the original data. The results of fitting a linear model are shown below in Table 3.

$$\ln(\hat{W}_Q) = -378.29 + .57 * \ln(1-\rho) + 10.09 * \ln(\rho) + 101.14 * \ln(\mu) -$$
$$.57z + .06 * (\ln(1-\rho))^2 + 11.61 * (\ln(\rho))^2 - 6.62 * (\ln(\mu))^2$$

Coefficients:

	Value	Std. Error	t value	Pr(>|t|)
(Intercept)	-378.2936	71.1325	-5.3182	0.0000
Log.mu.	101.1422	18.9918	5.3256	0.0000
Log.rho.	10.0855	2.1173	4.7635	0.0000
log.1.rho.	0.5687	0.2347	2.4229	0.0160
z.X2.1	-0.5664	0.0146	-38.8762	0.0000
I(Log.rho.^2)	11.6112	4.0001	2.9027	0.0040
I(log.1.rho.^2)	0.0564	0.0240	2.3509	0.0194
I(Log.mu.^2)	-6.6181	1.2632	-5.2392	0.0000

Residual standard error: 0.07313 on 292 degrees of freedom
Multiple R-Squared: 0.9628

F-statistic: 1081 on 7 and 292 degrees of freedom, the p-value is 0

Table 3. Transformed Mean Regression Results for Significant Terms

By replacing the reparameterization of variables used in the regression model with the original design factors, we get the following result.

$$\ln(\hat{W}_Q) = -378.29 + .57 * \ln(1 - .0000059\frac{x_3(x_1+x_4)}{x_2}) +$$
$$10.09 * \ln(.0000059\frac{x_3(x_1+x_4)}{x_2}) + 101.14 * \ln(60863.997\frac{x_2}{x_1+x_4}) +$$
$$.57z + .06 * (\ln(1 - .0000059\frac{x_3(x_1+x_4)}{x_2}))^2 +$$
$$11.61 * (\ln(.0000059\frac{x_3(x_1+x_4)}{x_2}))^2 - 6.62 * (\ln(60863.997\frac{x_2}{x_1+x_4}))^2$$

This model has a multiple R-Squared value of 0.9628 meaning that approximately 96 percent of the variability of the response variable is explained by the predictor variables in the model. While the R^2 for this model is slightly less than for the prior model, it has better predictive power in the region of low loss prediction.

B. STANDARD DEVIATION REGRESSION ANALYSIS

The second of the two metamodels is a second-order polynomial regression model for the standard deviation of the mean response that includes main effects and interaction terms.

$$Y = Std\ Dev(avg\ delay\ in\ queue)$$
$$E[Y] = \beta_0 + \sum_{j=1}^{5}\beta_j x_j + \sum_{i=j}^{5}\ \sum_{j=1}^{5}\beta_{ij}x_i x_j$$

$\hat{\beta}_0$ is the estimated regression coefficient for the intercept term. The estimated regression coefficients for the main effects are indicated by $\hat{\beta}_j$ for j=1,2,3,4,5. The estimated regression coefficients for the interaction and quadratic terms are indicated by $\hat{\beta}_{ij}$ for i,j=1,2,3,4,5. The following are the predictor variables for this model.

X_1=Length of reset phase

X_2=Percentage to overfill the unit

X_3=Target value for accessions

X_4=Length of the life cycle

X_5=Categorical variable for $X_2 > 1.0$

I used all 340 design points in my attempt to fit a polynomial regression function for the standard deviation of the average delay in queue for the system. Figure 11 depicts a scatter plot of the predictor variables versus the response variable.

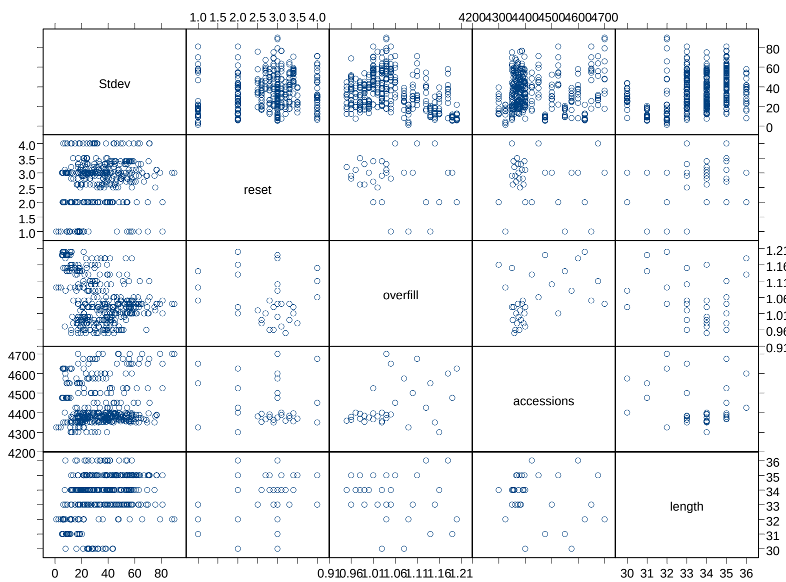

Figure 11. Scatter Plots of the Predictor Variables versus the Standard Deviation of the Delay in Queue

The relationships of the response to the overfill and length variables appear to be nonlinear. I fit a polynomial regression equation with all main effects, two-way interactions and quadratic effects. I removed those terms with a p-value less than 0.05. The results of a polynomial regression function are shown in Table 4 below.

$$\hat{s}_{W_Q} = -94.29 + 11.48x_1 + 3.23x_4 + 25.12x_5 - 1.98x_1^2 - 90.26x_2^2 + .0002x_3^2$$

```
Coefficients:
                     Value Std. Error  t value Pr(>|t|)
    (Intercept) -94.2935  26.3023    -3.5850   0.0004
          reset  11.4832   5.0492     2.2743   0.0236
         length   3.2273   0.5168     6.2448   0.0000
             x5  25.1169   2.1516    11.6735   0.0000
     I(reset^2)  -1.9780   0.9871    -2.0039   0.0459
  I(overfill^2) -90.2636   6.6782   -13.5162   0.0000
I(accessions^2)   0.0000   0.0000     5.2617   0.0000

Residual standard error: 13.36 on 333 degrees of freedom
Multiple R-Squared: 0.4625

F-statistic: 47.75 on 6 and 333 degrees of freedom, the p-value is 0
```

Table 4. Standard Deviation Regression Results for Significant Terms

A plot of the residuals versus the fitted regression equation shown in Figure 12 reveals heteroscedasticity.

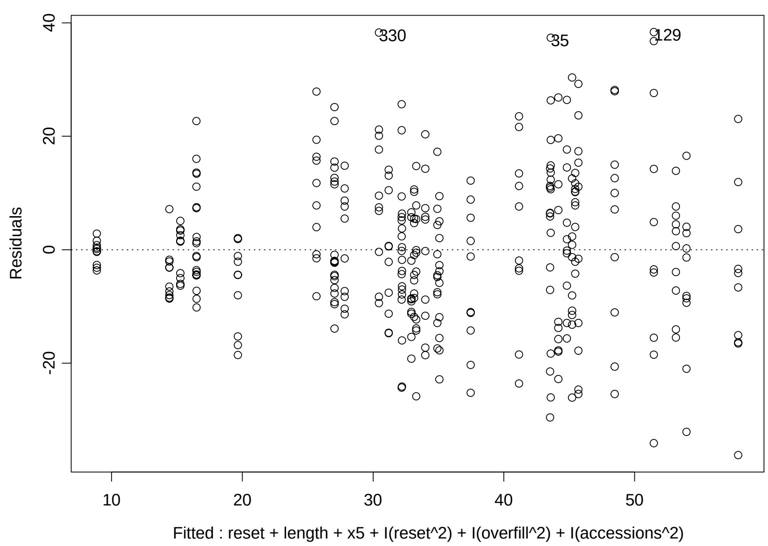

Figure 12. Plot of the Residuals versus the Fitted Equation

As previously mentioned, this polynomial regression function had poor predictive power when tested. Returning to the theory of queueing models, I reparameterized the design factors in terms of λ and μ. Since the distribution of W_Q is geometric, I began with the variance

of W_Q for an M/M/1 queueing model $\sigma_{W_Q}^2 = \frac{\rho}{(1-\rho)^2}$ (Ross, 2000). Therefore the variance of W_Q for a G/G/k queueing model is approximately $\sigma_{W_Q}^2 \approx k\frac{\rho}{(1-\rho)^2}$. The rest of the derivation is shown below.

$$\sigma_{W_Q}^2 \approx k\frac{\rho}{(1-\rho)^2} = k\frac{\lambda}{\mu}(1-\frac{\lambda}{\mu})^{-2}$$
$$= k\frac{\lambda}{\mu}(\frac{1}{\mu})^{-2}(\mu-\lambda)^{-2} = k\lambda\mu(\mu-\lambda)^{-2}$$

Taking the natural logarithm of this result transforms the product of parameters into a summation.

$$2\ln(\sigma_{W_Q}) = k_1\ln(\lambda) + k_2\ln(\mu) - 2k_3\ln(\mu-\lambda)$$

I used this result as a basis of a regression model for the standard deviation of the average delay in queue for this system in terms of λ and μ. I also included a categorical variable z representing values of X_2 that are greater than 1.0 as in the first metamodel. This model was fit with the data that excluded the design points with a value of $\rho > 1$. The results of this model are shown in Table 5 below.

$$\ln(\hat{s}_{W_Q}) = 9.5401 + 9.5840\ln(\lambda) - 10.4569\ln(\mu) + .5240z + .0102(2\ln(\mu-\lambda))^2$$

Replacing the reparameterization of the predictor variables with the original design factors gives the following result.

$$\ln(\hat{s}_{W_Q}) = 9.54 + 9.58\ln(.3627x_3) - 10.46\ln(60863.997\frac{x_2}{x_1+x_4}) -$$
$$.52x_5 + .01(2\ln(60863.997\frac{x_2}{x_1+x_4} - .3627x_3))^2$$

Coefficients:

	Value	Std. Error	t value	Pr(>|t|)
(Intercept)	9.5401	8.4140	1.1338	0.2578
Log.lambda.	9.5840	1.5171	6.3173	0.0000
Log.mu.	-10.4569	1.0815	-9.6690	0.0000
I(X2Log.mu.lambda.^2)	0.0102	0.0024	4.2817	0.0000
z.X2.1	0.5240	0.0773	6.7804	0.0000

Residual standard error: 0.4642 on 295 degrees of freedom
Multiple R-Squared: 0.521

F-statistic: 80.21 on 4 and 295 degrees of freedom, the p-value is 0

Table 5. Transformed Standard Deviation Regression Results

The R-Squared value of 0.521 indicates that approximately 52 percent of the variability is explained by the predictor variables in this model.

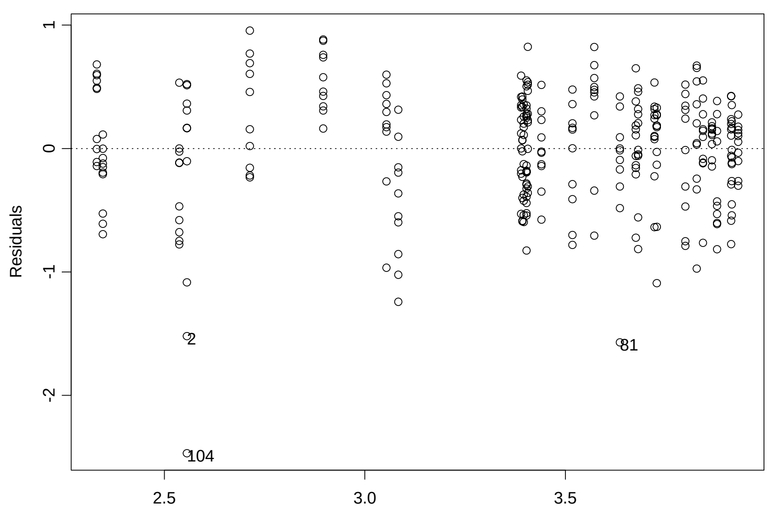

Figure 13. Plot of Residuals versus the Fitted Values

C. LOSS FUNCTION DEVELOPMENT

The loss function serves as a way to measure the performance of the system by examining the variability as well as the mean performance. Recall from chapter 3 that the expected loss is $E[Loss]=(\sigma_Y^2+(\mu_Y-\tau)^2)$. The mean performance and the variability are estimated using the two

metamodels constructed in the previous two sections. The resulting predicted loss function is shown below.

$$\hat{L}(x) = ((e^{\substack{9.5401+9.5840\ln(\lambda)-10.4569\ln(\mu)+.5240z+ \\ .0102(2\ln(\mu-\lambda))^2}})^2 + (e^{\substack{-378.2936+.5687*\ln(1-\rho)+10.0855*\ln(\rho)+ \\ 101.1422*\ln(\mu)-.5664z+.0564*(\ln(1-\rho))^2+ \\ 11.6112*(\ln(\rho))^2-6.6181*(\ln(\mu))^2}} - \tau)^2)$$

It is unrealistic to expect all lieutenants to have zero wait time before an assignment to a unit of action. Several lieutenants will have an opportunity to attend military schools such as Airborne School, Air Assault School and Ranger School upon completion of the OBC. Others may take leave upon graduation from OBC or choose to snowbird or blackbird. “Snowbirding” is an unofficial term for arriving early to a unit or training site (Hovda, 2002). “Blackbirding” is an unofficial term for remaining at a training site after graduation for a temporary period of time prior to receiving assignment orders (Hovda, 2002). A realistic target value for lieutenants to wait prior to unit assignment would be 30 days.

Having set the value for τ, the objective now was to find the optimal values of the factors in the model that provide the best solution. I took the partial derivatives of the loss function with respect to each variable, set the individual equations equal to zero and solved the system of equations for the five unknown variables that minimized the loss function. Using the Maple 9.5 software (Maplesoft, 2004), I derived a solution to minimize this loss function and found the optimal values for the five factors in the model. The values for the solution derived by Maple were outside the ranges for each of the factors. Therefore I took the appropriate upper or lower bound for the value of each factor. The five optimal values are listed below.

$$x_1 = 1.0, x_2 = 1.2, x_3 = 4300, x_4 = 30, x_5 = 1$$

I also used the solver tool in Excel as a second check and arrived at the same optimal design point to minimize the loss equation. The predicted loss at this optimal solution is 7948.457 days^2. The mean performance is 119.06 days with a variance of 17.389 days^2. I conducted my simulation again at this optimal solution for ten replications per noise factor for a total of 30 trials and averaged the delay in queue across the noise space. The results are listed in Table 6.

Trial	Average Delay in Queue (days)
1	114.56
2	116.23
3	114.65
4	114.52
5	111.63
6	113.56
7	117.98
8	119.59
9	116.86
10	119.05

Table 6. Results of Simulation at Optimal Values

The grand mean for these thirty trials is 115.86 days with a variance of 6.42 days^2. Substituting these values

into the loss equation yields an observed value of 7378.799 days^2.

E[Loss] = (Variance + (Mean - τ)^2

This is only slightly lower than the predicted loss of 7948.457 days^2. A 95 percent confidence interval for the mean and the standard deviation values are shown below.

Mean:

$$\hat{W}_Q \pm t_{\alpha/2,n-1}\hat{s}_{W_Q} / \sqrt{n}$$
$$119.06 \pm (1.649)(4.17/\sqrt{300})$$
$$119.06 \pm 0.397$$
$$[118.7, 119.5]$$

Stdev:

$$\hat{s}_{W_Q} \pm t_{\alpha/2,n-1} s / \sqrt{n}$$
$$4.170 \pm (1.649)(.4642/\sqrt{300})$$
$$4.170 \pm .0442$$
$$[4.126, 4.214]$$

These confidence intervals do not contain the mean and standard deviation of the response variable from the simulation runs at the optimal values. However, our objective is to find a recommendation where the expected delays are small. These results indicate a region of the response surface in which we may achieve a small expected delay with greater consistency.

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

One of the most significant insights gained during this process related to the stability of the system. The proposed life-cycle manning strategy was modeled as a queueing system with the arrival rate λ determined by the number of officer accessions per year and the service rate μ determined by the number of life-cycle units and their authorized strengths. The traffic intensity is the ratio of the arrival rate and the service rate, λ/μ. If the arrival rate is greater than the service rate, resulting in traffic intensity greater than 1, queuing systems become unstable and the queue grows without bound. This simulation terminated at ten years with very large wait times. Otherwise the average wait time for lieutenants would have grown to infinity.

Another reason queues develop in a system is because of variability. The number of officer accessions each year is a random variable and the schedule for these accessions varies from year to year. In order to reduce the length of the queues and therefore the wait times for officers, the schedule of officer accessions should be synchronized with the life-cycle of the units to reduce the variability of the matching process.

B. RECOMMENDATIONS

The objective of this analysis was to find a recommendation where expected delays for lieutenants are small. However, we are willing to trade off some expected delay in order to achieve greater consistency. This analysis indicates a region of the factor space that

provides the combination of smallest delay and consistency. I recommend shortening the length of the reset phase from 90 days to 60 days and the overall life-cycle length from 36 months to 32 months. In keeping with the theory of queueing models, this corresponds to higher service rates which will assist to reduce the wait time in the queue.

I further recommend reducing the annual accessions as much as can be allowed. The range of accessions considered in this analysis included values as low as 4300 officer accessions per year. This essentially reduces the arrival rate of the queueing model which also assists in minimizing the expected delay. Overfilling the life-cycle units to as much as 120 percent of the authorized strength is also recommended. In essence, this increases the service capacity by employing lieutenants who would otherwise be waiting.

A final recommendation is to synchronize the OBC schedules for each of the branches with the life cycles of the units. This synchronization of schedules will assist in eliminating variability in trying to pair new OBC graduates with life-cycle units.

C. FURTHER RESEARCH

The scope of this thesis only considered the impact of lieutenants waiting for life-cycle unit assignment. This analysis could be expanded to include all company-grade officers. The computer simulation ends with officer promotions from lieutenant to captain, but could be extended to include the Captain's Career Course and future assignments for captains. The queues that developed for lieutenants under the life-cycle manning strategy will most likely develop for captains.

This analysis used historical OBC schedules from the Army Training Requirements and Resources System (ATRRS) to plan future OBC schedules and graduation dates. However, implementation of life-cycle manning may create additional training requirements that alter this scheduling. Further research in this area could include analysis that determines an optimal schedule for synchronizing OBC graduations with the life cycles of units.

A final suggestion for further research would be calculating the scaling factor used in the predicted loss function to estimate that actual cost in dollars to the Army for implementing this manning strategy. This would enable decision-makers to directly compare the costs of delays in assignments to other costs in the overall manpower process.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. ACC BRANCHES AFFECTED BY LIFE-CYCLE MANNING

The following is a list of the Army Competitive Category branches that will be affected by the life-cycle manning strategy.

Table 7. 11 ACC Branches.

Infantry
Field Artillery
Armor
Engineers
Military Intelligence
Military Police
Quartermaster
Ordnance
Transportation
Signal
Chemical

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Buss, Arnold. 2001. *Basic Event Graph Modeling*. Simulation News Europe , Issue 31, pp. 1-6.

Buss, Arnold. 1996. *Modeling with Event Graphs.* In 1996 Winter Simulation Conference, pp. 153-160.

Buss, Arnold H., Paul J. Sanchez. 2002. *Building Complex Models with LEGOS (Listener Event Graph Objects)*. In 2002 Winter Simulation Conference, pp. 732-737.

Cioppa, T. M. 2002. *Efficient Nearly Orthogonal and Space-Filling Experimental Designs for High-Dimensional Complex Models.* Ph.D. Dissertation, U.S. Naval Postgraduate School, Monterey, California.

Deputy Chief of Plans, Human Resources Command. *Force Stabilization Leader Information Briefing*. Internet on-line. Available from <https://www.stabilization.army.mil/>. [November, 2004].

Elton, LTG (Retired) Robert M. 2002. *A Unit Manning System for the Objective Force*. Internet on-line. Available from <https://www.stabilization.army.mil/Historical%20Perspective/manning_only.htm/>. [November, 2004].

Hovda, Erik K. 2002. *A Simulation to Determine the Effect That the Army Basic Officer Leadership Course will have on Accession Training*. Master's Thesis, Naval Postgraduate School, Monterey, California.

Insightful Corporation. 2003. *S-PLUS 6.2 for Windows*, Seattle, WA.

Kleijnen, J. P., S. M. Sanchez, T. W. Lucas, and T. M. Cioppa. 2005. *A User's Guide to the Brave New World of Simulation Experiments*. INFORMS Journal on Computing, forthcoming

Law, Averill M., and W. D. Kelton. 2000. *Simulation Modeling and Analysis*, 3rd ed. The McGraw-Hill Companies, Inc.

Maplesoft, a Division of Waterloo Maple Inc. 2004. *Maple 9.50*.

Neter, John, Michael H. Kutner, Christopher J. Nachtsheim, and William Wasserman. 1996. *Applied Linear Statistical Models,* 4th ed. The McGraw-Hill Companies, Inc.

Ross, Sheldon M. 2000. *Introduction to Probability Models,* 7th ed. Academic Press.

Sanchez, S. M., P. J. Sanchez, J. S. Ramberg, and F. Moeeni. 1996. *Effective Engineering Design through Simulation*. International Transactions on Operational Research, Vol.3, No. 2, pp. 169-185.

SAS Institute Inc. 2003. *JMP User's Guide, Version 5.1*. Cary, North Carolina.

U.S., Department of the Army. 1986. *Army Regulation 600-83, The New Manning System – COHORT Unit Replacement System*.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. LTC Mark Lukens
   Deputy Chief of Staff, Army G-1
   Washington, D.C.

4. MAJ Erik Hovda
   Deputy Chief of Staff, Army G-1
   Washington, D.C.

5. Professor Paul Sanchez
   Naval Postgraduate School
   Monterey, California

6. Professor Samuel Buttrey
   Naval Postgraduate School
   Monterey, California

7. MAJ William Lewis
   Student, Naval Postgraduate School
   Monterey, California